STAR Online Software

D. Olson (leader), E. Yee
15 Nov. 1996

Subsystem summary

The STAR online software provides the top level control and bookkeeping functions for the experiment. The online software together with slow controls (interface to hardware), trigger, data acquisition and detector subsystems comprise the complete online system.

The online software provides a control model of loosely coupled state machines. This control model accounts for the synchronization and sequencing needs of the subsystems, where required, and also permits independent and asynchronous activities by the subsystems when inter-subsystem synchronization is not required. This model is implemented using a central serviced called the state manager.

The online software also provides a configuration management function for naming, archiving and restoring the detailed configurations of the complete online system. Details of the inter-subsystem dependencies (like daq-trigger interface) are coordinated by high-level names. Subsystems keep configuration details grouped by these names and the detailed interface between two subsystems are kept consistent based on the name of the configuration.

Additional components of the online software are:

- The operator user interface for run control and monitoring
- A sequencer that drives the control model by sending state transition events to the subsystems which is able to read scripts that describe the proper sequencing.
- A central alarm service including alarm handling and user interface.
- A central message logging service including filtering functions and user interface.

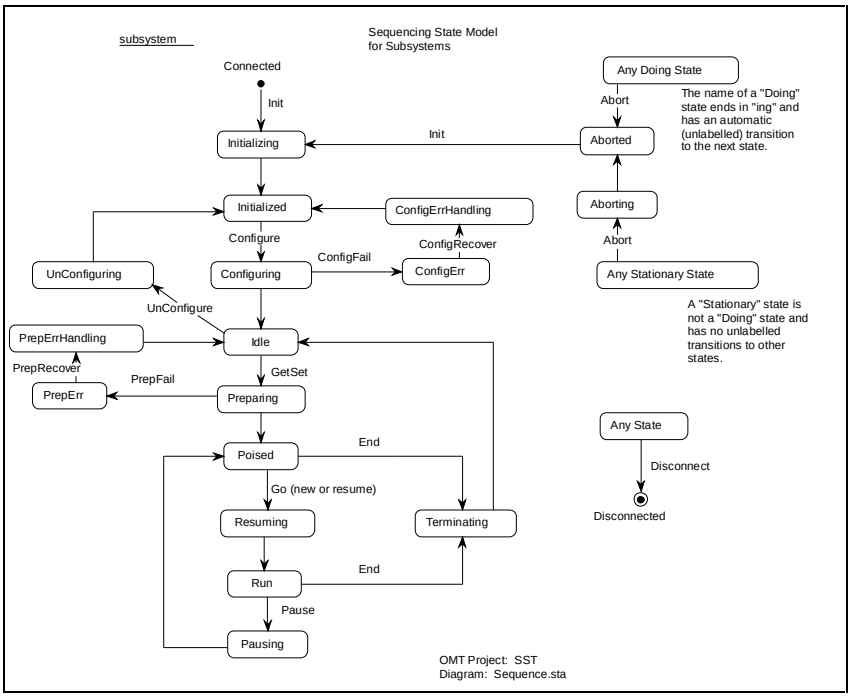
FY96 status

The major accomplishments in FY96 are the implementation of a state manager, the development of a sequencing state transition model, and the development of a first prototype run control interface used with the STAR system test.

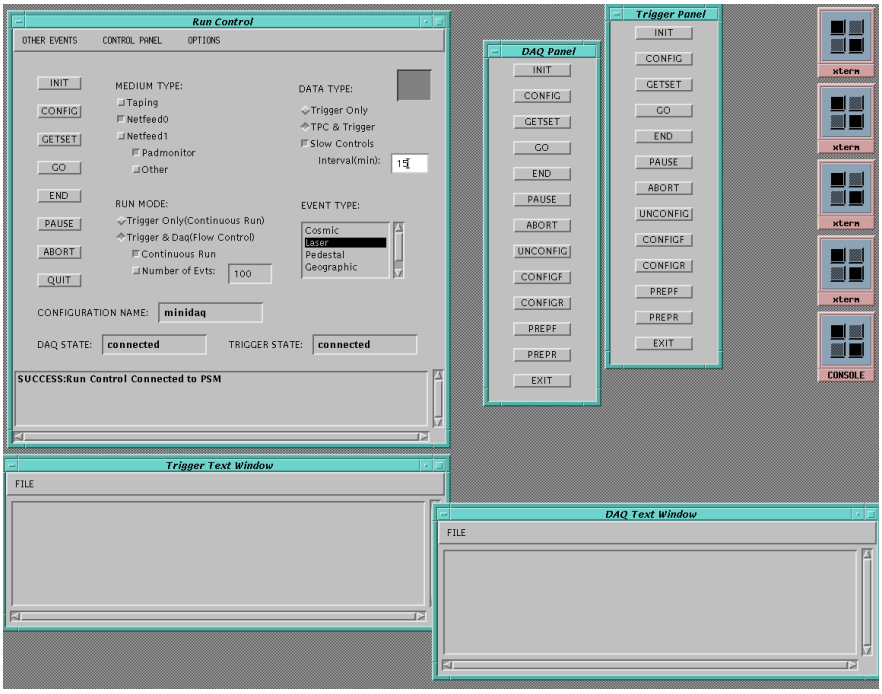
The state manager is implemented as a central server that uses ASCII messages over TCP/IP. It provides a client API (in C on VxWorks, SunOS, Solaris, WinNT/95) that guaranties the internal consistency of the allowed state transition model. The state transition events which are sent to subsystems have additional information carried along as arguments to the events. This argument mechanism is used to tell subsystems the name of the configuration which should be used.

The state transition model (see figure on next page) used by this state manager was developed from initial ideas and review of state models used by other experiments (at CERN and TJNAF). It has evolved from the initial model based upon experience with prototypes and detailed discussions with members of several subsystems.

A first prototype of a run control interface has been implemented (in Java) for use with the state manager in the STAR system test (see figure on next page). This initial implementation serves two primary functions, the first being a necessary function for the system test and cosmic ray test of the TPC, and secondly as a basis of experience upon which people can refine their ideas about requirements and functionality for the real run control to be used at BNL.



A schematic diagram of the sequencing state transition model currently being used in the STAR system test. An example startup sequence is: (Connected)-Init->(Initializing)-(Initialized)-Configure->(Configuring)-(Idle). An example data acquisition run sequence is: (Idle)- GetSet->(Preparing)-(Poised)-Go->(Resuming)-(Run)-End->(Terminating)-(Idle).



A screen view of the prototype run control interface for the STAR system test. The main window (Run Control) shows configuration selections and action buttons which correspond to sequencer scripts, as well as some monitoring information. Message windows for both trigger and daq are shown below. Individual windows (at top right) allow sending individual state transition events to daq and trigger.